DIGITAL DIFFERENTIAL ANALYZER DATA SYNCHRONIZER

FIELD OF THE INVENTION

The present invention relates to a method and apparatus for the synchronization of data between two distinct clock domains.

BACKGROUND OF THE INVENTION

In numerous systems capable of the transmission and the reception of data, there is often a need for the synchronization of data transmitted with an associated clock signal having known frequency and varying phase, with the clock signal of the receiving system. For example, such synchronization is necessary in a master system which transmits a request data stream with an associated request clock signal to a subsidiary system which receives the data stream and request clock signal and transmits a response data stream, with an associated response clock signal. The master system receives the response data steam and associated response clock signal. The response clock signal has a known frequency, but unknown phase, due to a variable delay in the data and clock paths.

The variable delay occurs in both the request data stream and request clock signal and the response data stream and response clock signal due to the combination of circuit delays and wire transmission delays. The delay may also vary after the system is initialized, due to the dependence of the delays upon, for example, temperature, operating voltage, and physical tension of the cable connecting the systems. In both cases (i.e., transmission and reception), the received clock signal can be used to generate a sampling clock which is used to reliably sample the data. The subsidiary system may use the sampling clock to synchronously clock all its storage elements or delay elements (such as latches, flip-flops, and phase delays), thus avoiding any problem of clock asynchrony.

However, the master system is faced with a Hobson's choice, as the master system may choose synchrony with either the request clock signal or the response clock signal, but not both. If synchrony with the request clock signal is maintained, a data synchronizer must be placed in the receiver of the master system, as the phase of the response clock signal is not predictable with respect to the request clock signal.

The choice of an optimal main system clock rate may depend on factors other than the required communication rate between the main and subsidiary systems, such as the delay of internal circuits and interconnections, required internal computation rates, and so on. When the rate of the main system clock is different from the rate of the request clock signal, both the request clock signal and response clock signal are asynchronous with the main system clock, and data synchronization is required in data paths both in the request data path and the response data path.

Thus, in systems transmitting and receiving data to and from various subsidiary systems, there exists a need for a method and apparatus for synchronizing both request data streams and response data streams to the clock signal of the receiving device, which accomplishes the synchronization with minimal delay and maximum reliability.

SUMMARY OF THE INVENTION

Accordingly, it is an object of the present invention to provide a method and apparatus for synchronizing a request data signal (also referred to as a data stream) to a communications clock and to synchronize a response data signal to a main system clock, with the synchronization being accomplished with minimal delay and maximum reliability.

More specifically, the present invention relates to a method of synchronizing a data stream between two distinct clock domains. The novel method comprising the steps of receiving the data stream at a first clock rate M, sequentially loading the data stream into a plurality of registers at the first clock rate, utilizing a digital differential analyzer to generate a synchronization signal having a frequency proportional to a ratio of the first clock rate M and a second clock rate N, and sequentially reading the plurality of registers at a rate corresponding to the frequency of the synchronization signal.

The present invention also relates to an apparatus for synchronizing a data stream between two distinct clock domains. The apparatus comprises an input means for receiving the data stream at a first clock rate M, means for sequentially loading the data stream into a plurality of registers at the first clock rate, a digital differential analyzer operative for generating a synchronization signal having a frequency proportional to a ratio of the first clock rate M and a second clock rate N; and means for sequentially reading the plurality of registers at a rate corresponding to the frequency of the synchronization signal.

As described in detail below, the method and apparatus of the present invention provides important advantages over the prior art. Most importantly, the present invention performs the required data synchronization with minimal delay and maximum reliability.

The present invention accommodates large variations in phase delay of the subsidiary system, variations which may be in excess of the cycle time. Additional delay is accommodated by adjusting the number of registers into which the data stream is loaded into and read from.

The present invention avoids the use of repeated asynchronous sampling of the high bandwidth signals and clocks, which is a source of failure in prior art synchronizer designs. Asynchronous sampling is the use of an unsynchronized clock to sample a signal or clock; when the sample point occurs nearly simultaneous with the time that the signal or clock changes, an inherent metastability occurs in the sampling circuit, which can take an arbitrarily long time to be resolved into a stable logic level. After initialization, the present invention has no use of asynchronous sampling, so it is free from synchronizer failure.

The present invention allows for the use of a wide variety of clock rates in a single design, as timing-sensitive delay elements are limited to the phase-locked loops and phase-delay elements. These elements are explicitly programmable with the desired clock rates (M and N) so that necessary adjustments to these elements need not be discovered dynamically from the incoming clock signals.

Additional advantages of the present invention will become apparent to those skilled in the art from the following detailed description of exemplary embodiments of the present invention.

The invention itself, together with further objects and advantages, can be better understood by reference to the following detailed description and the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 illustrates a data communication system having a main system and a subsidiary system and which incorporates an exemplary embodiment of a digital differential analyzer data synchronizer in accordance with the present invention.

Fig. 2 illustrates an exemplary embodiment of the request data synchronizer of the present invention.

Fig. 3 illustrates an exemplary embodiment of the response data synchronizer of the present invention.

Fig. 4 illustrates an exemplary embodiment of the digital differential analyzer shown in Figs. 2 and 3.

DETAILED DESCRIPTION

The following detailed description relates to a novel digital differential analyzer data synchronizer. In the description, numerous specific details are set forth, such as word size, clock rates etc., in order to provide a thorough understanding of the present invention. It will be obvious, however, to one skilled in the art that these specific details need not be employed to practice the present invention. In other instances, well-known system structures have not been described in detail in order to avoid unnecessarily obscuring the present invention.

As stated above, Fig. 1 illustrates an exemplary data communication system which incorporates the digital differential analyzer data synchronizer of the present invention. The system provides the ability to transfer data between systems operating with different clock references. In the embodiment of Fig. 1, the two systems are referred to as the main system 20 and the subsidiary system 40. It is noted that only the data transmission and reception sections of the "main" and "subsidiary" systems, which incorporate the digital differential analyzer data synchronizer of the present invention are illustrated.

As shown, the main system 20 receives a master clock 15 (also referred to as a reference clock) as an input signal. The reference clock 15 is coupled to two phase-locked loop "PLL" clock generators 21, 22, which are utilized to generate a communications clock 71 and a main system clock 72. Typically, the communications clock has a frequency of N times the reference clock frequency and the master clock has a frequency of M times the reference clock frequency, where N < M.

The main system 20 further comprises a request data synchronizer 23, a response data synchronizer 27, a reset generator 24, a multiplexer 25, a quadrature circuit 28 and two amplifiers 26 and 29.

Referring again to Fig. 1, the request data synchronizer 23 receives as inputs, the communications clock 71, the main system clock 72, a reset signal generated by the reset generator 24 and the request data stream to be synchronized. The request data synchronizer 23 generates a synchronized request data stream 31 and a request data strobe 32 as outputs. While the operation of the request data synchronizer 23 is explained in detail below, generally speaking, it functions to synchronize the request data stream (i.e., the data transferred to the subsidiary system) to the communications clock 71.

Fig. 2 illustrates an exemplary embodiment of the request data synchronizer 23 of the present invention. As shown, the request data synchronizer 23 comprises a first 5-bit rotate register 41 which is clocked by the main system clock 72 and loaded such that only a single bit is equal to "1" (e.g., "10000"). The rotate register 41 is enabled to rotate only on N of every M cycles by digital differential analyzer 45. The synchronizer 23 further comprises five 16-bit registers 42 each of which: (1) receives the request data stream as an input; (2) is clocked by the main system clock 72; and (3) are enabled by the first rotate register 41. The output of each of the five 16-bit registers 42 is coupled to the input port of a 5:1 multiplexer 43. The output of the multiplexer 43 is selected by the output of a second 5-bit rotate register 44, which is clocked by the communications clock 71 and loaded such that only a single bit is equal to "1". The synchronizer 23 further comprises a digital differential analyzer 45 which receives N and M as inputs, and which is enabled by the main system clock 72. The digital differential analyzer "DDA" is a sequential circuit which generates an enabling output with frequency y/x relative to its input clock. The use of a DDA is well-known as a technique for scan-converting lines for a raster display, also referred to as Bresenham's Line Algorithm. (See J.D. Foley and A. Van Dam, Fundamentals of Computer Graphics, Addison-Wesley, 1982, pp 432-436.)

The operation of the request data synchronizer 23 is as follows. Initialization is critical to proper operation of the data synchronizer. Data loaded into the data registers 42 must not be read until it is certain that the data is stable; delays occur due to the inherent delay of the register storage element itself and known variations in the relative delay of the two clock signals. The number of data registers 42 and initial state of the first rotate register 41 and the second rotate register 44 is selected to ensure that a sufficient delay occurs between loading data into a data register 42 and reading the data, so that data is always present and stable when read. Because the reset signal is asynchronous to at least one of the main system clock 72 and the communications clock 71, there can be a small uncertainty in the initial state of the rotate registers, this is dealt with by selecting a sufficient initial state of the rotate registers to ensure reliable operation regardless of position of the reset signal.

The first rotate register 41 functions to sequentially select one of the five 16-bit registers to receive the 16-bit request data stream (i.e., loaded into the register). As both the first rotate register 41 and each of the five 16-bit registers 42 are clocked by the main system clock, and are enabled by the DDA 45, on each enabled clock cycle of the main system clock, a new value of the request data stream (i.e., a 16-bit word) is loaded into one of the five registers 42. The loading of registers 42 occurs in a circular manner, meaning that after the fifth register 42 has been loaded, the rotate register 41 enables the first register 42 again and a new value is written into the first register. Thus, the loading process continually repeats itself.

As indicated above, the second rotate register 44 operates in combination with the multiplexer 43 to select one of the five registers 42. Importantly, however, the second rotate register 44 is clocked by the communications clock 71. The DDA 45 receives N and M as inputs, and functions to output N enabling signal values every M clock cycles (the output signal of the DDA 45 is referred to herein as the synchronization signal).

As a result, the synchronized request data stream output by the synchronizer 23 comprises N words of data every M clock cycles, where each word of data represents the output of a single register 42. In addition, the DDA 45 functions to spread the N words of data as evenly as possible over M clock cycles. As such, the request data synchronizer 23 synchronizes the request data stream, which was read into the synchronizer 23 at the main system clock rate, to the communications clock 71.

It is noted that in the foregoing embodiment of the request data synchronizer 23, with regard to sequentially reading registers 42 via the multiplexer 43 and the rotate register 44, it must be accomplished such that the data read into the registers 42 has sufficient time to stabilize. This is accomplished by initializing the two rotate registers with their "1" bit in offset locations, as shown by register 41 initialized to "10000" and register 44 to "00100".

After the request data stream is synchronized to the communications clock 71 the stream is coupled to multiplexer 25. Multiplexer 25 is controlled by the communications clock 71 and functions to transmit the data stream on the rising and falling edges of the clock signal. In the current embodiment, during each clock cycle of the communications clock 71, a 16-bit word of the synchronized request data stream is presented to the multiplexer 25, along with an additional two bits 37 which represent the state of the communications clock 71 (i.e., either high or low). Signal 37 causes the multiplexer 25 to generate a copy of the communications clock 71 as one of the 9 bits transmitted via amplifier 26, but this copy is designed to have the same timing characteristics as the synchronized request data stream. In particular, the rising and falling times of the generated copy of the communications clock matches that of the 8 data signals of the synchronized request data stream, a fact used by the subsidiary system to receive and recover the data. On a rising edge of the communications clock 71, a first 8-bits of the 16-bit data word present at the multiplexer 25 and one bit indicating that the communications clock 71 is high, are transmitted to the subsidiary system 50 via amplifier 26. Then, on the next falling edge of the communications clock 71, the remaining 8-bits of the word present at the multiplexer 25, and an additional bit indicating that the communications clock 71 is low, are transmitted to the subsidiary system 50. As such, 8-

bits of data are transmitted to the subsidiary system every half-cycle of the communications clock 71.

The subsidiary system 50 comprises an amplifier 51 and quadrature circuit 51, which operate to receive the transmitted signal and reconstruct the 16-bit word of the synchronized request data stream. As shown, the quadrature circuit 51 functions to separate the 8-bit portion and the single bit representing the communications clock 71 upon receipt of the 9-bit signal. The bit representing the communications clock 71 is coupled to a phase shifter 53. The phase shifter 53 generates a clock signal with the same clock rate as the communications clock 71 with an appropriate phase for sampling the data at a time when the data is stable, approximately a 90 degree (one-quarter cycle) phase shift. The clock signal generated by the phase shifter 53 is then utilized to sample the data stored in the registers 54 and 55 on both the rising edge and the falling edge of the signal output by the phase shifter 53. As a result, the 16-bit data word is recovered each complete clock cycle of the communications clock 71.

The subsidiary system further comprises a multiplexer 56 and an amplifier 57, which function in an identical manner to the multiplexer 25 and amplifier 26 of the main system 20, to transmit 8 bits of a 16-bit data word of the response data stream every half-cycle of the communications clock 71. As shown, the multiplexer 56 is controlled by the output of the phase shifter

53, and receives a 16-bit data word and a two bit signal 58, which is essentially a copy of the signal output by the phase shifter 53.

Referring again to Fig. 1, the main system 20 comprises an amplifier 29 and a quadrature circuit 28, which is identical to the quadrature circuit 52 of the subsidiary system 50, and functions reconstruct the 16-bit response data stream transmitted by the subsidiary system 50.

The reconstructed response data stream is then coupled to the response data synchronizer 27, which is structurally similar to and functions in the a similar manner as the request data synchronizer 23. Specifically, the response data synchronizer 27, transforms the response data stream, which as received, is at the same rate but asynchronous with the communications clock 71, to a data stream which is synchronous with the main system clock (the operation of the request data synchronizer 23 is only different from the response data synchronizer 27 in that the incoming data stream is transformed from the main system clock domain to the communications clock domain). The subsidiary system generates a variable phase delay to the response data stream, therefore the precise phase of the received clock is not as precisely known as the communications clock 71, and consequently, the response data synchronizer may require a greater number of instances of data register 42 than the number of instances required in the request data synchronizer. To clarify the similarity of the request and response data synchronizer, the same number of instances (five) of the data register 42 are shown in each.

Referring to Fig. 3, which is illustrates an exemplary embodiment of the response data synchronizer 27 of the present invention, the response data synchronizer 27 receives as inputs, the communications clock 71, the main system clock 72, a reset signal generated by the reset generator 24 and the response data stream to be synchronized. The response data synchronizer 27 generates a synchronized response data stream 48 and a request data strobe 49 as outputs.

As with the request data synchronizer 23, the response data synchronizer 27 comprises a first 5-bit "rotate" register 41 which is clocked by the communications clock 71 and loaded such that only a single bit is equal to "1". The synchronizer 27 also comprises five 16-bit registers each of which: (1) receive the response data stream as an input, (2) are clocked by the communications clock 71, and (3) are enabled by the first rotate register 41. The output of each of the five 16-bit registers 42 is coupled to the input port of a 5:1 multiplexer 43. The output of the multiplier 43 is controlled/selected by the output of a second 5-bit "rotate" register 44, which is clocked by the main system clock 72 and loaded such that only a single bit is equal to "1". The synchronizer 27 further comprises a digital differential analyzer 45 which receives the communications clock

71 and the main system clock 72 as inputs, and which is enabled by the main system clock 72.

Upon initialization, the first rotate register 41 functions to sequentially select one of the five 16-bit registers to receive the 16-bit request data stream (i.e., loaded into the register). As both the first rotate register 41 and each of the five 16-bit registers 42 are clocked by the communications clock 71, on each clock cycle of the communications clock 71, a new value of the response data stream is loaded into one of the five registers 42. The loading of registers 42 occurs in a circular manner, meaning that after the fifth register 42 has been loaded, the rotate register once again enables the first register 42 and a new value is written into the first register, and the loading process begins again.

As indicated above, the second rotate register 44 operates in combination with the multiplexer 43 to select one of the five registers 42. The second rotate register 44 is clocked by the main system clock 72 and enabled by the output of the DDA 45. The DDA 45 receives N and M and main system clock 72 as inputs, and functions to output a synchronization signal having N pulses every M clock cycles. As a result, the synchronized response data stream output by the synchronizer 23 comprises N words of data every M clock cycles, where each word of data represents the output of a single register 42. In addition, the DDA 45 functions to spread the N words of data as evenly as possible over M clock cycles. As such, the response data synchronizer 27 functions to synchronize the response data stream to the main system clock 72.

Fig. 4 shows one embodiment of the digital differential analyzer shown in Figs. 2 and 3. As explained in detail below, this embodiment of the DDA computes the n/m frequency pattern once, at system reset, and then recycles the result through a rotating register. The advantage of this embodiment of the present invention is that the DDA implementation may operate at a much lower rate than the main system clock, which permits the use of slower, smaller, lower power technology for the DDA. The DDA may also be powered off after system reset, further reducing power.

Referring again to Fig. 4, as shown therein the DDA 45 comprises a modified DDA 61, which preferably comprises slower, smaller, lower power technology, a shift register 62, a rotate register 63 and a M-cycle delay circuit 64. The modified DDA receives a reset signal 65, a slow clock signal 66, and M and N as inputs.

In operation, the M-cycle delay 64 receives the reset signal 65 and delays the signal M cycles. The reset signal is delayed by M cycles because the DDA 61 requires M cycles to produce the pattern representing the synchronization signal (i.e., the relationship between the distinct clock domains). For example, the pattern may represent the even distribution of N pulses over a period of M cycles. As such, in the first M cycles after reset, the DDA 61 functions to fill the shift register 62 with the pattern of 1's and 0's representing the synchronization signal. After the initial delay period of M cycles, the delayed signal is transmitted to the rotate register 63, whereby the register 63 is loaded with the value in the shift register 62. Rotate register 63 also receives M as an input which operates to select the length of the rotate register, which is variable.

After loading the pattern into the rotate register 63 and defining a length, the rotate register 63 is clocked by a fast clock that corresponds to the main system clock, and repetitively produces the synchronization signal (i.e., data strobe 69) at a high speed. The data strobe 69 is coupled to the enable input of rotate register 44 illustrated, for example, in Fig. 2, and functions to control the reading of the registers 42.

It should also be apparent that the DDA 45 shown in figures 2 and 3 have the same inputs and produce the identical output, so that a single instance of the DDA 45 may be employed for both the request data synchronizer 23 and response data synchronizer 27. It should further be apparent that if the request and response data synchronizers have the same number of data registers 42, then the rotate register 41 of one data synchronizer and the rotate register 44 of the other data synchronizer have the same inputs and produce identical outputs, except for a rotation of the output bits. Accordingly, a single instance of these blocks may be employed for both the request data synchronizer 23 and response data synchronizer 27. It should further be apparent that when multiple subsidiary systems are employed as the same communications clock rate, that a single DDA 45 (and a single PLL) can be employed for each instance of the request and response synchronizers. It should also be apparent that multiple instances of the subsidiary system may be inserted in series without disturbing the basic characteristics of the synchronization system. It should also be apparent that any circuit which produces the necessary n/m pattern can be employed by the circuit illustrated in Fig. 4 to load the rotating register. For example, a PLA may be used to generate the pattern, given some limited-range values for N and M.

As stated the method and apparatus of the present invention provides important advantages over the prior art. Most importantly, the use of the present invention performs the required synchronization with minimal delay and maximum reliability. The present invention also provides for the use of a DDA utilizing slower, smaller, lower power technology, so as to further reduce cost and minimize necessary operating power requirements.

Although the elements of the present invention have been described in conjunction with an exemplary embodiment, it is appreciated that the invention may be implemented in a variety of other ways. Consequently, it is to be understood that the particular embodiment shown and described by way of illustration are in no way intended to be considered limiting. Reference to the details of these embodiments is not intended to limit the scope of the claims which themselves recite only those features regarded as essential to the invention.